CS206 Data Structure

Homework #3

Sungwon Cho

0. About Programs

Program Language: C# 5.0 (.NET 4.5)
Libraries: Standard library of .NET(ex: System, System.Core)
Requirements: OS that supports .NET Framework 4.5 or Mono

1. Telephone ADT

structure TelBook ▷ Telephone Book

objects: a finite ordered *group*s that connected.
group is object that finite ordered (number, name, gname)s that connected.
functions: $(\forall tbook \in TelBook, number \in \Sigma^+, name \in \Sigma^+, gname \in \Sigma^+)$

TelBook **Create**()
make new empty *TelBook* and **return** it

void **AddTel**(*tbook*, *number*, *name*, *gname*)
if (*number* already exist in *tbook*) **throw** exception
make new objects that store *number*, *name*, *gname*
if (*gname* group is not exist) **CreateGroup**(*tbook*, *gname*)
connect it after head in *gname* group

void **RemoveTel**(*tbook*, *number*)
find *number* in each existing groups and remove it if exists

void **Find**(*tbook*, *name*)
prints data which is *name* in each existing groups in *tbook*

void **ShowAll**(*tbook*)
prints all data in each existing groups in *tbook*

void **ShowGroup**(*tbook*, *gname*)
prints all data in *gname* group in *tbook*

void **CreateGroup**(*tbook*, *gname*)
if (*gname* group already exist in *tbook*) **throw** exception
make group that its name is *gname* and connect it after head in *tbook*

2. ADT Implemented with Singly Linked List

Source Files: Telephone.cs, STelNode.cs, SGroup.cs, STelBook.cs, Program1.cs
Instruction: Program.inst.txt
Sample Input, Output: Program.sample.txt

3. ADT Implemented with Doubly Linked List

Source Files: Telephone.cs, DTelNode.cs, DGroup.cs, DTelBook.cs, Program2.cs
Instruction: Program.inst.txt
Sample Input, Output: Program.sample.txt

4. Compare Singly vs Doubly Linked List

Before compare operations, assume that both linked already have k groups, and each group has n_i data where $\Sigma_{i=1}^{k} n_i = n$. Also, I only consider substitution and comparison as operation. Now, compare add operation of singly and doubly linked list. Below code is **AddTel** method and related methods of singly linked list.

```
public class STelBook {
  public void AddTel(string obj) {
    Telephone tel = (Telephone)obj;
    SGroup insgroup = null, group = head.Next;
    while (group != null) {
      if (group.ExistTel(tel.Number))
        throw new Exception("Telephone number already exist.");
      if (group.Name == tel.Group)
        insgroup = group;
      group = group.Next;
    }
    if (insgroup == null) {
      insgroup = new SGroup(tel.Group, head.Next);
      head.Next = insgroup;
    }
    insgroup.AddTel(tel);
  }
}

public class SGroup {
  public void AddTel(Telephone tel) {
    head.Next = new STelNode(tel, head.Next);
  }

  public bool ExistTel(string tel) {
    STelNode node = head.Next;
    while (node != null) {
      if (node.Tel.Number == tel)
        return true;
```

```
      node = node.Next;
    }
    return false;
  }
}
```

First, look at **AddTel** method in **SGroup**. It just performs 1 operation. In **ExistTel** method, 1 operation executed before while loop and $3n_i + 1$ operation executed in while loop at most(in case of such tel does not exist). Finally, look at **AddTel** method in **STelBook**. Worst case is that group which its name is gname does not exist. Before execution of while loop, 3 operation executed. In while loop, $(\Sigma_{i=1}^{k}(3n_i + 2) + 3) + 1$ operation executed. After while loop, 4 operation executed. Overall, $3 + (\Sigma_{i=1}^{k}(3n_i + 2) + 3) + 1 + 4 = 3n + 5k + 8$ operation executed. Since $k \leq n$, its time complexity is $O(n)$.

Below code is **AddTel** method for doubly linked list.

```
public class DTelBook {
  public void AddTel(string obj) {
    Telephone tel = (Telephone)obj;
    DGroup insgroup = null, group = head.Next;
    while (group != rear) {
      if (group.ExistTel(tel.Number))
        throw new Exception("Telephone number already exist.");
      if (group.Name == tel.Group)
        insgroup = group;
      group = group.Next;
    }
    if (insgroup == null) {
      insgroup = new DGroup(tel.Group, head, head.Next);
      head.Next.Prev = insgroup;
      head.Next = insgroup;
    }
    insgroup.AddTel(tel);
  }
}

public class DGroup {
  public void AddTel(Telephone tel) {
    DTelNode node = new DTelNode(tel, head, head.Next);
    head.Next.Prev = node;
    head.Next = node;
  }

  public bool ExistTel(string tel) {
    DTelNode node = head.Next;
    while (node != rear) {
      if (node.Tel.Number == tel)
        return true;
      node = node.Next;
```

```
    }
    return false;
  }
}
```

First, look at **AddTel** method in **DGroup**. It just performs 3 operation. In **ExistTel** method, 1 operation executed before while loop and $3n_i + 1$ operation executed in while loop at most(in case of such tel does not exist). Finally, look at **AddTel** method in **DTelBook**. Worst case is that group which its name is gname does not exist. Before execution of while loop, 3 operation executed. In while loop, $(\Sigma_{i=1}^{k}(3n_i + 2) + 3) + 1$ operation executed. After while loop, 5 operation executed. Overall, $3 + (\Sigma_{i=1}^{k}(3n_i + 2) + 3) + 1 + 5 = 3n + 5k + 9$ operation executed. Since $k \leq n$, its time complexity is $O(n)$.

Also, compare remove operation of singly and doubly linked list. Below code is **RemoveTel** method for singly linked list.

```
public class STelBook {
  public void RemoveTel(string tel) {
    SGroup group = head.Next;
    while (group != null) {
      if (group.RemoveTel(tel))
        return;
      group = group.Next;
    }
    throw new Exception("There's no number to be deleted.");
  }
}

public class SGroup {
  public bool RemoveTel(string number) {
    STelNode node = head;
    while (node.Next != null) {
      if (node.Next.Tel.Number == number) {
        node.Next = node.Next.Next;
        return true;
      }
      node = node.Next;
    }
    return false;
  }
}
```

First, look at **RemoveTel** method in **SGroup**. In this method, 1 operation executed before while loop and $3n_i+1$ operation executed in while loop at most(in case of such number does not exist). Finally, look at **RemoveTel** method in **STelBook**. In this method, 1 operation executed before while loop and $(\Sigma_{i=1}^{k}(3n_i+2)+2)+1$ at most(in case of such number does not exist). Overall, $1+(\Sigma_{i=1}^{k}(3n_i+2)+2)+1 = 3n+4k+2$ operation executed. Since $k \leq n$, its time complexity is $O(n)$.

Below code is **RemoveTel** method for doubly linked list.

```
public class DTelBook {
  public void RemoveTel(string tel) {
    DGroup group = head.Next;
    while (group != rear) {
        if (group.RemoveTel(tel))
          return;
        group = group.Next;
    }
    throw new Exception("There's no number to be deleted.");
  }
}

public class DGroup {
  public bool RemoveTel(string number) {
    DTelNode node = head.Next;
    while (node != rear) {
      if (node.Tel.Number == number) {
        node.Prev.Next = node.Next;
        node.Next.Prev = node.Prev;
        return true;
      }
      node = node.Next;
    }
    return false;
  }
}
```

First, look at **RemoveTel** method in **DGroup**. In this method, 1 operation executed before while loop and $3n_i+1$ operation executed in while loop at most(in case of such number does not exist). Finally, look at **RemoveTel** method in **DTelBook**. In this method, 1 operation executed before while loop and $(\Sigma_{i=1}^{k}(3n_i+2)+2)+1$ at most(in case of such number does not exist). Overall, $1+(\Sigma_{i=1}^{k}(3n_i+2)+2)+1 = 3n+4k+2$ operation executed. Since $k \leq n$, its time complexity is $O(n)$.

Overall, for both add and remove operations, difference of number of operation executed in singly and doubly linked list is less than 10. Time complexity(Big-O) is same as $O(n)$. This happens because there is no operation that need to get back of node. So, there are no need to use doubly linked list and waste computer memory.